REMARKS

This response is to the official action mailed in the above-referenced case on 01/24/2007, made Final. Claims 1-11 are standing for examination. Claims 1, 3-9 and 11 are rejected under 35 U.S.C. 102(b) as being anticipated by Batch, as taught by "Not a Batch Language: A Control Language!", E.H. Bristol, published May 1995, hereinafter Bristol. Claims 2 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bristol as applied to claims 1 and 8 above in view of UML as taught by Integrating UML Diagrams for Production Control Systems by Hans J. Kohler et al, ACM, 2000.

In response to the Examiner's rejections and objections, applicant herein amends claims 1 and 8 to more particularly point out the subject matter believed to be patentable by applicant. Applicant also presents extensive arguments clearly showing where the art of Batch fails to anticipate applicant's claims, as amended.

Applicant points out that Batch teaches a method of executing static batch programs including a plurality of procedure pages consisting of objects and definitions for running and controlling complex auto-startup in a multi-unit production plant while utilizing a uniform control language. Applicant argues that Batch is actually utilizing a control language controlling how the batch programs execute control over the plant. Applicant provides a method for actually creating or generating a batch program which does not extend further than the software program, itself. It is clear to the applicant that the Examiner is using some functions of controlling a production plant as in Batch, to read on applicant's claimed ability to generate code creating actual batch programs.

Batch programs need to process large volumes of data in a definite time window. Owing to the large volumes of data being processed, batch programs are resource intensive. If a batch program fails in the middle, it must recover and restart from the point of failure without having to redo the computations from the start. Thus, writing a batch program is a complex task needing wide ranging expertise. The framework of the present invention ensures that on failure a batch program will restart automatically, from a consistent state, with minimal loss of computation. The framework also provides automation support for management of resources such as memory.

The art of Bristol clearly teaches a process control paradigm called 'batch control' that is typically seen in chemical plants, manufacturing plants etc. Applicant's invention has nothing to do with industrial process control. Bristol teaches (pg. 8) The Language models a process hierarchically in terms of its Operations/Objects, modeling the process divisions: the Styrene Plant Operation and the Furnace, Reactor. Heat Recovery, Separator, and Feed Tankage SubOperations. Each Operation is organized into Pages for modeling distinct control functionalities. The example shows several of these Pages H mostly Procedures Pages, describing active control procedures.

Applicant's method of generating a batch includes viewing a batch program to comprise a 'fixed part' and a 'variable part'. All batch programs will share the same fixed part and each specific batch program will have a different variable part. Applicant's invention encapsulates the fixed part in the form of a framework with placeholders where the variable parts can be plugged in. In the art of Bristol, the framework is the plant operations, therefore, one could not apply the software of Bristol to any other operation without extensive manual programming. Applicant's invention provides a high level mechanism for specifying application-specific variable parts from which their implementations can be automatically generated.

Applicant's claim 1 provides means for generating a batch program through a plurality of abstractions, each representing a batch program; a batch function of the program; operation of the function; a data provider to the function; and an abstraction representing a context class of the function. Clearly, the procedural pages taught in the art of Bristol fail to read on the generation of hierarchical abstractions as claimed in applicant's invention. Applicant's invention also includes a code library which is not necessary in the art of Bristol because Bristol fails to teach actually generating batch programs or code as claimed.

Applicant believes independent claims 1 and 8 are therefore patentable as amended and argued above. Dependent claims 2-7 and 9-11 are therefore patentable on their own merits, or at least as depended from a patentable claim.

If there are any time extensions required for response in addition to any extension petitioned and paid with this response, such petition is requested, and if there are any fees due over any fees paid with this response, authorization is given to deduct the fees from deposit account 50-0534.

Respectfully submitted,
Vinay Vasant Kulkarni et al.

By /Donald R. Boys/
Donald R. Boys
Reg. No. 35,074

Central Coast Patent Agency, Inc.
3 Hangar Way, Suite D
Watsonville, CA 95076
831-768-1755